

# PM Controls: Merlin Visuals

---

- ◆ All controls have new visuals (except value set/)
- ◆ Design objectives:
  - improve visual appearance of existing PM controls
  - no new styles or APIs required to use new look
  - existing applications using controls inherit new appearance
  - no regressions in existing applications
- ◆ Other new visual enhancements:
  - new notebook style - the tabbed dialog
  - new color enhancements
  - new default font
  - close button on title bars

# PM Controls: New Notebook

---

- ◆ New styles:
  - **BKS\_TABBEDDIALOG**
  - **BKS\_BUTTONAREA**
- ◆ New messages:
  - **BKM\_SETTABCOLOR**
  - **BKM\_SETNOTEBOOKBUTTONS**
- ◆ New push button style:
  - **BS\_NOTEBOOKBUTTON**
- ◆ New push button message:
  - **BM\_AUTOSIZE**

# PM Controls: New Color APIs

---

- ◆ Controls obtain colors from following sources (in descending order of preference):
  - presentation parameter
  - application control color (*new*)
  - global control color (*new*)
  - default system color
- ◆ Control colors are set and queried via new APIs (user interface available).
- ◆ All PM controls respond to new control colors, will not interfere with existing color schemes.
- ◆ New presentation parameters created for some controls.

# PM Controls: New Color APIs

---

- ◆ Each control has its own *control color type*:
  - » CCT\_NOTEBOOK
  - » CCT\_SCROLLBAR
  - » CCT\_STATICTEXT
  - » CCT\_GROUPBOX
- ◆ Each color used by a control type has a *control color index*:
  - » CCI\_BACKGROUND
  - » CCI\_FOREGROUND
  - » CCI\_BORDERLIGHT
  - » CCI\_BORDERDARK
- ◆ Not all color indexes are used by all color types
  - e.g. CCT\_STATICTEXT does not use CCI\_BORDERLIGHT

# PM Controls: New Color APIs

---

- ◆ New color APIs:
  - **WinQueryControlColors**
  - **WinSetControlColors**
- ◆ New color messages:
  - **WM\_QUERYCTLTYPE**
  - **WM\_CTLCOLORCHANGE**

# PM Controls: Other New Features

---

## ◆ Close button:

- added automatically to windows which already have a min/max menu on title bar.
- if window does not have min/max menu, can use **FCF\_CLOSEBUTTON** to add it.
- removed from titlebar if window does not have enabled “Close” item in the system menu.
- system menu notifies frame whenever the state of the “Close” is changed.

## ◆ New default font: **9.WarpSans**

- used in default scheme for menu and titlebar.
- not a feature of new controls - controls use whichever font is used in application or Workplace Shell.

# PM Controls: Debugging Visuals

---

- ◆ New painting code replaced old - no major interface redesign - mostly rewritten functions
- ◆ Most bitmaps replaced by drawing code - allow colors to be changed
- ◆ MAGNIFY is a useful tool to allow you to view visuals close up
- ◆ If you fix a visual defect, make sure that all views of the control are working:
  - e.g. for listbox, combinations of the following:
    - » normal
    - » owner-draw
    - » variable height owner-draw
    - » right-to-left (bidi)
    - » borders/no borders
    - » horizontal scroll on/off

# PM Controls: Debugging Notebook

---

- ◆ Old notebook style should work exactly as in W 3.0 - no differences
- ◆ New notebook style piggy-backed on top of old
  - no minor tabs - only code for major tabs has been added
  - most of the new code follows either:
    - » `if (pBookInstance->ulNotebookStyles & BKS_TABBEDDIALOG`
    - » `if (pBookInstance->ulNotebookStyles & BKS_BUTTONAREA)`
  - new tab painting functions in **bookdtab.c**
- ◆ Most bugs probably deal with painting tabs correctly
  - important functions which set up data for painting:
    - » **ResetVisPagePtrs**
    - » **ResetTabData**
  - best place to start in most cases:
    - » **GotoPage**

# PM Controls: Color Internals

---

- ◆ All presentation parameters and system colors s work as before where applicable
  - **Note:** PP\_BORDERCOLOR is not used in as many places
- ◆ New internal function for querying color in a co
  - **CtlQueryColor** replaces **QueryPresColors**
  - **CtlQueryColor** called by control when it is created, and w color change has occurred.
  - **CtlQueryColor** uses tables set up in **initterm.c** to keep tra which color should be used where.

# PM Controls: Color Internals

---

- ◆ Each color used in controls has unique identifier
  - a constant in **controlp.h**, for example:
    - » **CCV\_LISTBOXBGND** : list box background color
    - » **CCV\_LISTBOXFGND** : list box foreground (text) color
    - » **CCV\_LISTBOXBDRDLIGHT** : light outer list box border
    - » **CCV\_LISTBOXBDRDARK** : dark outer list box border
    - » **CCV\_COMBOBOXBGND** : combo box background color
    - » **CCV\_COMBOBOXFGND** : combo box foreground color
    - » **etc.**
  - each constant is used in the controls to identify which color need to query, and maps to an entry in the master table:

```
plbox->clrHiliteFore = CtlQueryColor(hwnd, hps, CCV_LISTBOXHILITEFGND, 0);
plbox->clrHiliteBack = CtlQueryColor(hwnd, hps, CCV_LISTBOXHILITEBGND, 0);
plbox->clrTopOuterBorder = CtlQueryColor(hwnd, hps, CCV_LISTBOXBDRDARK, 0);
plbox->clrTopInnerBorder = CtlQueryColor(hwnd, hps, CCV_LISTBOXBDR2DARK, 0);
plbox->clrBottomOuterBorder = CtlQueryColor(hwnd, hps, CCV_LISTBOXBDRDLIGHT, 0);
plbox->clrBottomInnerBorder = CtlQueryColor(hwnd, hps, CCV_LISTBOXBDR2LIGHT, 0);
```

# PM Controls: Color Internals

---

- ◆ Master table of color usage is **vCCmap** in **init**
- ◆ Contains entries for all colors used in all controls
  - Four columns:
    1. **CCI\_** index                      control color index
    2. **PP\_** index                        presentation parameter
    3. **QPF\_** flags                      e.g. **QPF\_INHERIT**
    4. **SYSCLR\_** index                  system color
  - or
  - RGB\_** value                      RGB color if system color available.
- ◆ **vCCmap** used to build global and per-process color tables:
  - **vCCglobal**                      global default colors
  - **vCCapp**                          per process default colors

# PM Controls: Color Internals

---

- ◆ **vCCapp** entries initialized to -1 which means no process default set - use entry in **vCCglobal**
- ◆ Examples:
  - **Push button background** (**CCV\_PUSHBUTTONBGND**)
    - » No presentation parameter
    - » No entry in **vCCapp** (-1)
    - » Entry in **vCCglobal** set to **SYSCLR\_BUTTONMIDDLE**  
Push button will paint with **SYSCLR\_BUTTONMIDDLE**
  - **Scroll bar inner border** (**CCV\_SCROLLBARBDR2LIG**)
    - » Owner has **PP\_BORDER2LIGHTCOLOR** set to **RGB\_RED**
    - » **vCCapp** contains **RGB\_GREEN**
    - » **vCCglobal** contains **RGB\_YELLOW**  
Scroll bar inner border will paint with **RGB\_RED** color.

# PM Controls: Debugging Colors

---

## ◆ Potential problems:

- incorrect colors are used
  - » an entry in **vCCmap** is wrong
- in 256 color mode, palette manager ruins colors in color set
  - » customer has set colors which are not in the default 256 set

## ◆ Where to start:

- set breakpoint on **CtlQueryColor** and trace through to find where the color value is coming from.

# PM Controls: Potential APARs

---

## ◆ Potential Merlin APARs with new visuals:

### – clipping:

» most controls have wider borders (2 pels vs. 1 pel) reducing space available inside control. Affects controls with text

◆ push button

◆ list box / combo box

◆ entry field

◆ spin button

### – emulated controls:

» some apps. create controls which look like system controls (e.g. Lotus 123/G menus). User has no way of knowing if it is an application or OS/2 bug

### – visual inconsistency:

» possible that some apps will not look as good with new visuals

# PM Controls: Potential APARs

---

## ◆ Other potential APARs:

### – close button:

- » Does not work well with Lotus 123/G and Freelance (certain versions). Patch available on system (LOTUSFIX.CMD).

### – new default font (9.WarpSans):

- » *not* a system font - will not appear in application dialog boxes, set by application or user.
- » Can be lost when playing around with schemes and preferences notebooks.
- » Old device drivers replace DSPRES.DLL which will delete the new WarpSans font from system.
- » Controls in new settings notebooks may be clipped in certain resolutions (800x600, 1280x1024, etc). WPS layout preferences notebook.

# PM Controls: Color Internals

---

Control Color Index -----	CCT_STATICTEXT -----	CCT_GROUPBOX -----	CCT_BUTTON -----
CCI_FOREGROUND	Y	Y	Y
CCI_FOREGROUNDREADONLY			
CCI_BACKGROUND	Y	Y	Y
CCI_BACKGROUNDDIALOG	Y	Y	Y
CCI_BORDER			
CCI_BORDERLIGHT		Y	Y
CCI_BORDERDARK		Y	Y
CCI_BORDER2			
CCI_BORDER2LIGHT			
CCI_BORDER2DARK			
CCI_BORDERDEFAULT			
.			
.			
.			

# PM Controls: Color Internals

---

Control Color Index

CCT\_STATICTEXT

CCT\_PUSHB

-----

-----

-----

CCI\_FOREGROUND

CCV\_STATICTEXTFGND

CCV\_PUSHB

CCI\_FOREGROUNDREADONLY

CCI\_BACKGROUND

CCV\_STATICTEXTBGND

CCV\_PUSHB

CCI\_BACKGROUNDDIALOG

CCV\_STATICTEXTBGNDDLG

CCV\_PUSHB

CCI\_BORDER

CCI\_BORDERLIGHT

CCV\_PUSHB

CCI\_BORDERDARK

CCV\_PUSHB

CCI\_BORDER2

CCI\_BORDER2LIGHT

CCI\_BORDER2DARK

CCI\_BORDERDEFAULT

CCV\_PUSHB

.

.

.

# PM Controls: Color Internals

---

CCV_ Constants -----	Control Color Index -----	Presentation Parameter -----	Presp Flags -----
/* CCV_PUSHBUTTONBGND	*/ CCI_BACKGROUND,	PP_BACKGROUND_COLOR,	QPF_NOINHERIT
/* CCV_PUSHBUTTONFGND	*/ CCI_FOREGROUND,	PP_FOREGROUND_COLOR,	QPF_NOINHERIT
/* CCV_PUSHBUTTONDISABLEDBGND	*/ CCI_DISABLEDBACKGROUND,	PP_DISABLEDBACKGROUND_COLOR,	QPF_NOINHERIT
/* CCV_PUSHBUTTONDISABLEDFGND	*/ CCI_DISABLEDFOREGROUND,	PP_DISABLEDFOREGROUND_COLOR,	QPF_NOINHERIT
/* CCV_PUSHBUTTONDEFAULTBDR	*/ CCI_BORDERDEFAULT,	PP_BORDERDEFAULT_COLOR,	0,
/* CCV_PUSHBUTTONBDRLIGHT	*/ CCI_BORDERLIGHT,	PP_BORDERLIGHT_COLOR,	0,
/* CCV_PUSHBUTTONBDRDARK	*/ CCI_BORDERDARK,	PP_BORDERDARK_COLOR,	0,
// CCV_PUSHBUTTONFIRST			
// CCV_PUSHBUTTONLAST			

# **PM Controls Enhancements**

---

**Mike Walker - Visuals**

**Tom Bellwood - DAX**

**Merle Sterling - Container**